

1

基于深度学习来生成图的论文

Sheng Xiang, Dong Wen, Dawei Cheng, Ying Zhang, Lu Qin, Zhengping Qian, Xuemin Lin: General graph generators: experiments, analyses, and improvements

1:对基于深度学习的通用图形生成器的概述

2:通用图形生成器的系统性和全面的性能比较

3:没有算法可以在图形模拟质量和效率（可扩展性）之间实现良好的权衡

4:对于通用图形生成器的用户来说，没有方便的工具包

在章节2，我们正式定义了问题，并概述了本文评估的一般图生成器。然后，我们详细介绍了图生成器，并在Sect中介绍了它们的优化目标。

在第章节4，我们对通用图形生成器进行了一些改进。

通用图形生成器的综合实验结果以及我们开发的工具包的介绍在 Sect.5. 第 6 节结束本文。

第二节定义的问题

问题陈述。给定一组观察到的图形 $\{G\}$ ，一个通用的图生成器旨在学习一个生成模型来捕获图的结构分布，使得一组新的图 $\{G'\}$ 可以生成具有相似结构分布。

可供改进的地方:

论文中提到了另一篇论文Erdős, P., Rényi, A.: On random graphs i. publicationes mathematicae (debrecen). 1959

介绍了基于 E-R 的图生成器的朴素实现为每对顶点生成概率为 p 的边，时间复杂度为 $O(n^2)$ 和空间复杂度 $O(m+n)$ 。显然这个构造算法可能会导致生成的图不能连通，可用于二分图构造那一节的对比材料

论文提到了Akoglu, L., Faloutsos, C.: RTG: a recursive realistic graph generator using random typing[J]. Data Min. Knowl. Discov. **19**(2):194–209 (2009)

四个参数 (K, W, q, β) 来控制生成分布。 K 是一个单词中可能的字符数， W 是单词数。 q 是键入空格的概率。 β 控制在 2D 键盘上随机键入对角线字符和非对角线字符的概率。这些参数有助于生成具有幂律度分布和社区的图形。此外，RTG 还可以生成二分图。基于 RTG 的图形生成器的实现需要 $O(m \log n)$ 时间复杂度和 $O(m+n+K^2)$ 空间复杂性。

2

同样是基于深度学习的构造算法

Bacciu, D., Micheli, A., Podda, M.: Graph generation by sequential edge prediction. ESANN (2019)

使用机器学习模型生成图形是一个具有挑战性的问题，在各个研究领域的应用中都有应用。在这里，我们提出一个基于循环深度学习的模型来生成图形学习预测其有序边缘序列。尽管它很简单，但我们在各种数据集上的实验表明，我们的方法能够生成源自非常不同分布的图，优于图论中的规范图生成模型，并达到性能可与图形生成方面的当前技术水平相媲美。

3

同样是基于深度学习的构造算法

Edge-based sequential graph generation with recurrent neural networks

我们使用这个顺序过程来设计一个新的生成模型，该模型由两个学习预测图边缘的递归神经网络组成：第一个网络生成每个边缘的一个端点，而第二个网络生成另一个端点，条件是第一个网络的状态。

4

暂时还没看懂

Bagan, G., Bonifati, A., Ciucanu, R., Fletcher, G.H.L., Lemay, A., Advokaat, N.: gmark,: Schema-driven generation of graphs and queries. IEEE TKDE 856–869,(2017)

在本文中，我们介绍了gMark的设计和工程原理，gMark是一个独立于域和查询语言的图形实例和查询工作负载生成器。gMark 的一个核心贡献是它能够定位和控制生成的实例和与这些实例耦合的生成的工作负载的属性多样性。其他新颖性包括支持常规路径查询、基本图形查询范例和模式驱动的查询选择性估计，这是控制工作负载阻塞点的关键功能。

我们将图生成的问题形式化，查询工作负载生成，并显示它们通常很难处理（第 3 节）

我们提供gMark的深入介绍设计原则，用于生成图形（第 4 节）和查询工作负载（第 5 节）。最值得注意的是新颖的功能是支持递归查询和查询生成的查询工作负载中的选择性估计

我们凭经验展示了 gMark 覆盖的能力多样化的图形和查询工作负载，准确性估计的选择性和生成器的可扩展性（第6节）。

我们提出了一个深入的实验比较使用gMark的最先进的图形查询引擎的代表性选择，揭示了当前图形查询处理引擎的重要局限性，特别是递归查询处理（第7节）。

5

似乎是一个类似于图生图的技术？

Barrett, C. L., Beckman, R. J., Khan, M., Kumar, V. S. A., Marathe, M. V., Stretz, P. E. , Dutta, T., Lewis, B. L.: Generation and analysis of large synthetic social contact networks. In WSC, pages 1003–1014. IEEE, 2009、

摘要:我们描述了基于“第一原则”的方法，用于开发综合城市和国家规模的社会联系网络。与简单的随机图技术不同，这些方法使用现实世界的的数据源，并将它们与行为和社会理论相结合来合成网络。我们为美国开发了一个合成人口，对人口中的每个人进行建模，包括家庭结构、人口统计和 24 小时活动序列。该过程涉及收集和操作集成到数据交换通用架构中的公共和专有数据集，然后使用这些数据集生成新的关系。社会接触网络源自基于互动人员物理共址的合成人口。我们使用图形度量来比较和对比跨越不同城市地区的社交网络的结构特征。然后，我们模拟这些网络上的扩散过程，并分析网络结构的异同。

介绍了诸多构造随机图的算法

Batagelj, V., Brandes, U.: Efficient generation of large random networks. Phys. Rev. E, page 036113, 2005

摘要:例如, 经常生成随机网络, 以研究模型参数对网络属性的影响或测试算法的性能。最近对大规模网络统计的兴趣引发了对可以快速生成大量大型网络的网络生成器的需求不断增长。我们在这里提出了简单有效的算法, 根据最常用的模型随机生成网络。它们的运行时间和空间要求在生成的网络大小上是线性的, 并且很容易实现。

A. $G(n, p)$ 给每条边给定一个数值 r , 如果 $r < p$, 就生成这条边。显然是 n^2 复杂度

因此, 我们使用 [9] 的几何方法来跳过未创建的潜在边(仅寻找未被创建的边?)。

计算出期望要进行实验的次数 k , 然后跳过 k 条边, 连边

ALG. 1: $\mathcal{G}(n, p)$

Input: number of vertices n , edge probability $0 < p < 1$

Output: $G = (\{0, \dots, n-1\}, E) \in \mathcal{G}(n, p)$

$E \leftarrow \emptyset$

$v \leftarrow 1; \quad w \leftarrow -1$

while $v < n$ **do**

 draw $r \in [0, 1)$ uniformly at random

$w \leftarrow w + 1 + \left\lfloor \frac{\log(1-r)}{\log(1-p)} \right\rfloor$

while $w \geq v$ **and** $v < n$ **do**

$w \leftarrow w - v; v \leftarrow v + 1$

if $v < n$ **then** $E \leftarrow E \cup \{v, w\}$

算法2本质上相当于随机选边, 显然仅适合于稀疏图

ALG. 2: $\mathcal{G}(n, m)$ in linear expected time

Input: number of vertices n
number of edges $0 \leq m \leq \binom{n}{2}$
Output: $G = (\{0, \dots, n-1\}, E) \in \mathcal{G}(n, m)$

$E \leftarrow \emptyset$
for $i = 0, \dots, m-1$ **do**
 repeat
 draw $r \in \{0, \dots, \binom{n}{2} - 1\}$ uniformly at random
 until $e_r \notin E$
 $E \leftarrow E \cup \{e_r\}$

对算法2的一个优化算法，构造一个替换数组，来实现已构造边对未构造边的映射，加速构造。

ALG. 3: $\mathcal{G}(n, m)$ in linear worst-case time

Input: number of vertices n
number of edges $0 \leq m \leq \binom{n}{2}$
Output: $G = (\{0, \dots, n-1\}, E) \in \mathcal{G}(n, m)$

$E \leftarrow \emptyset$
for $i = 0, \dots, m-1$ **do**
 draw $r \in \{i, \dots, \binom{n}{2} - 1\}$ uniformly at random
 if $e_r \notin E$ **then**
 $E \leftarrow E \cup \{e_r\}$
 else $E \leftarrow E \cup \{e_{\text{replace}[e_r]}\}$
 if $e_i \notin E$ **then**
 $\text{replace}[e_r] \leftarrow i$
 else $\text{replace}[e_r] \leftarrow \text{replace}[e_i]$

f
r
i
g
t
s
c
t
c

c
r
c
s

后续的一些算法是小世界模型算法和优先依附算法，

我在connectedpapers上找到了一些相关的文章

7

同样是基于深度学习的构造算法

Bojchevski, A., Shchur, O., Zügner, D., Günnemann, S.: Netgan: Generating graphs via random walks. In: ICML, pp. 610–619, 2018

我们提出了NetGAN——第一个能够模仿真实世界网络的图的隐式生成模型。我们将图生成问题作为学习输入图上偏向随机游走的分布。所提出的模型基于随机神经网络，该网络生成离散输出样本，并使用Wasserstein GAN目标进行训练。NetGAN能够生成表现出众所周知的网络模式的图形，而无需在模型定义中明确指定它们。同时，我们的模型表现出很强的泛化特性，正如其竞争性链接预测性能所突出的那样，尽管没有专门为此任务进行训练。作为结合这两种理想特性的第一种方法，NetGAN为进一步研究开辟了令人兴奋的途径。

8

综述型论文

Bonifati, A., Holubová, I., Prat-Pérez, A., Sakr, S.: Graph generators: State of the art and open challenges. ACM Comput Surv 53(2):1–30 (2020)

大量互连数据推动了图形生成器的设计和实现，这些生成器再现了现实世界的链接属性，或衡量了操纵这些数据的图形算法、技术和应用程序的有效性。我们考虑跨多个字段的图生成，例如语义网、图数据库、社交网络和社区检测，以及通用图。尽管这些社区对现代图形生成器的要求不同，但我们在一个共同的保护伞下分析它们，触及功能、实际用途及其支持的操作。我们认为，这种分类满足了为科学家、研究人员和从业者提供合适的数据生成器的需求。本调查通过关注那些与多个数据密集型任务相关且适合的图形生成器，全面概述了最先进的图形生成器。最后，我们讨论了当前图形生成器的开放挑战和缺失需求，以及它们未来对新兴领域的扩展。

9

同样是基于深度学习的构造算法

Brockschmidt, M., Allamanis, M., Gaunt, A.L., Polozov, O.: Generative code modeling with graphs. ICLR. !OpenReview.net (2019)

源代码的生成模型是一个有趣的结构化预测问题，需要对硬句法和语义约束以及自然的、可能的程序进行推理。我们为这个问题提出了一个新的模型，它使用图来表示生成的输出的中间状态。生成过程将语法驱动的扩展步骤与图形增强和神经消息传递步骤交织在一起。实验评估表明，我们的新模型可以生成语义上有意义的表达式，优于一系列强基线。

总结

除了Efficient generation of large random networks这篇介绍了一个优化生成E-R随机图的算法，这个算法我在<https://dl.acm.org/doi/abs/10.1145/1951365.1951406> 找到了进一步扩充的论文，有一定的对比参考意义。其他的论文基本介绍的是基于机器学习的图数据生成方法。

此前搜图构造算法的论文一直用的graph construct 导致一直没找到相关论文，搜索Graph Generation后找到了不少

有一篇论文:Fast Random Cactus Graph Generation https://link.springer.com/chapter/10.1007/978-3-030-10448-1_12#citeas

很适合作为仙人掌那一节的对比

下面是那篇论文的单线程算法:

1:生成一个连通图

2:用dfs, 将连通图转为树

3.1:随机一个对节点(x,y)

3.2:验证, 如果(x,y)将(x,y)加入图后, 图仍然是仙人掌, 就添加边(x,y)

检验的方法是:执行一个从 x 到 y 的遍历路径, 判断是否存在属于 a 的边 (a, b)循环;**(为了随机性牺牲了太多的复杂度)**

算法复杂度是 $O(N^2)$

多线程算法就是分块, 将节点sqrt地分块后分别构造, 再连接起来**(并行算法又失去了随机性)**

改进思路

指明论文中图生成算法是根据图的直接性质构造, 具有稳定的正确性与确切的复杂度, 且易于实现。指出与现有的深度学习图生成算法的区别

找几个论文作为对比, 比如上面提到的Fast Random Cactus Graph Generation

指出优势区间:稳定的正确性, 单独用于构造一类的图,

给论文中的构造算法找出一个明确的应用场景。